

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-63576

(43)公開日 平成10年(1998)3月6日.

(51)Int.Cl. <sup>6</sup>	識別記号	弁内整理番号	F I	技術表示箇所
G 0 6 F 12/08		7623-5B	G 0 6 F 12/08	F
	3 2 0	7623-5B		3 2 0
	3/06		3/06	3 0 2 A
	5 4 0			5 4 0
	12/16	7623-5B	12/16	3 2 0 D

審査請求 未請求 請求項の数4 OL (全 10 頁)

(21)出願番号 特願平8-224792

(22)出願日 平成8年(1996)8月27日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 高本 良史

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 小川 勝男

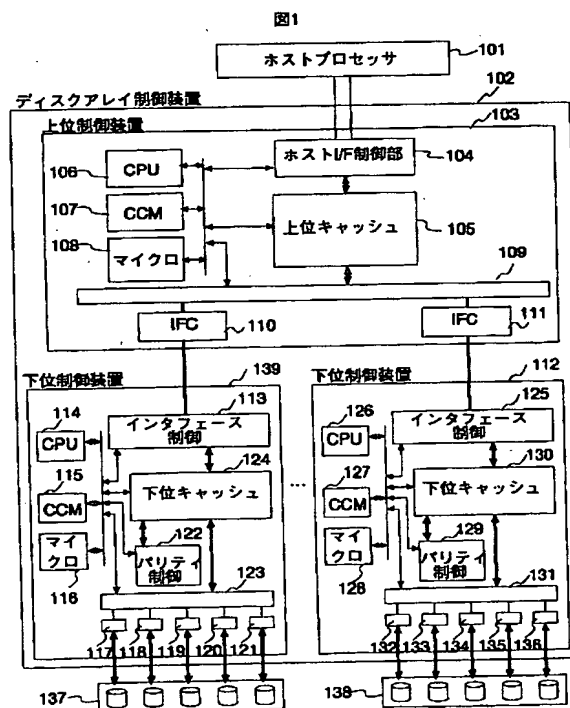
(54) 【発明の名称】 階層ディスク装置およびその制御方法

(57) 【要約】

【課題】 階層化されたディスクキャッシュにおいて、各キャッシュ間でデータの重複が発生しキャッシュの利用効率が低下するのを抑える。

【解決手段】 階層化された制御装置のそれぞれに、自制御装置の階層レベルを格納するメモリを設け、データの読み出し時に前記メモリ内の前記階層レベルを読み出し、最上位制御装置でなければ自キャッシュにはデータを格納しない制御を行う。

【効果】 各キャッシュ間で格納されているデータの重複が発生しないため、キャッシュの利用効率が向上する。



## 【特許請求の範囲】

【請求項1】ホストプロセッサに接続された最上位制御装置と、前記最上位制御装置に接続された複数の下位制御装置と、前記複数の下位制御装置に接続された複数のディスク装置と、前記最上位制御装置および前記複数の下位制御装置に設けられたディスクキャッシュからなる階層ディスク装置であって、

前記下位制御装置は、前記ホストプロセッサからの読み込みデータを、ディスクキャッシュに格納すること無く前記上位制御装置へ転送する手段と、

前記最上位制御装置は、前記読み込みデータを、ディスクキャッシュに格納し、かつ、前記ホストプロセッサへ転送する手段を有する階層ディスク装置。

【請求項2】複数のディスク装置を並列に動作させるとともに、冗長データを格納するディスクアレイ制御装置であって、ホストプロセッサに接続された最上位制御装置と、前記最上位制御装置に接続された複数の下位制御装置と、前記複数の下位制御装置に接続された複数のディスク装置と、前記最上位制御装置および前記複数の下位制御装置に設けられたディスクキャッシュからなる階層ディスク装置において、

前記ホストプロセッサからのデータ読み込み時に前記階層レベルが最上位制御装置かどうかを、前記最上位制御装置および前記複数の下位制御装置が有する階層レベルを示す識別子から判断し、

最上位制御装置でなければ前記読み込みデータをディスクキャッシュに格納しない制御を行うことを特徴とする階層ディスクキャッシュ制御方法。

【請求項3】前記ホストプロセッサからデータの書き込み要求が発生した時、冗長データを生成する制御装置は、上位制御装置に対して最新のデータが存在するかどうか検索し、

存在すれば上位制御装置のディスクキャッシュから下位制御装置のディスクキャッシュに前記データを転送することを特徴とする請求項2記載の階層ディスクキャッシュ制御方法。

【請求項4】前記最上位制御装置のキャッシュが障害を起こした場合、前記最上位制御装置に接続された複数の下位制御装置の前記識別子を、最上位制御装置を示す識別子に変更することを特徴とする請求項2記載の階層ディスクキャッシュ制御方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ディスク制御装置に係り、階層化されたディスクキャッシュの利用効率を高め性能を向上させるの階層ディスク制御装置およびその制御方法に関する。

## 【0002】

【従来の技術】一般的にコンピュータシステムは、プロセッサと2次記憶装置から構成されている。主として使

用される2次記憶装置は磁気ディスク装置である。現在、ディスク記憶装置の容量の伸び率は極めて高いが、メカニカルな動作を伴う磁気ディスク装置の性能はプロセッサ性能の伸び率ほど高くない。その課題を解決する方法として、以下の2つの方法が知られている。

【0003】第一の方法は、ディスク制御装置内にディスクキャッシュを設ける方法である。ディスクキャッシュは、一度読み込まれたデータを半導体メモリであるディスクキャッシュに格納しておき、同一のデータに対して再度読み込み要求が発生したときにはディスクキャッシュからデータを取り出すことで、ディスク装置へのアクセスを削減し高速化を行う方法である。キャッシュを増やすほどヒット率が向上するため性能も向上するが、容量当たりのコストが磁気ディスクに比べ高価であるため、通常は磁気ディスク容量の数10分の1から数100分の1の容量が設けられている。キャッシュ制御方法は、最も頻繁にアクセスされる順に格納しておき、最もアクセスされなかったデータの追い出しを行う。これをLRU処理といい、キャッシュ制御方法では一般的である。これらの方法に関して、特開平6-231045号公報に記載がある。

【0004】第二の方法は、磁気ディスクを並列に制御することで性能を向上させる方式である。D. Patterson, G. Gibson, and R. H. Kartzらによる“A Case for Redundant Arrays of Inexpensive Disks (RAID), in ACM SIGMOD Conference, Chicago, IL”, PP. 109-116 (June 1988) (以下、第1の参考文献と呼ぶ)では、複数のディスクドライブにデータを分散して配置することでディスク内に格納されたデータへのアクセス時間を短縮し、かつパリティあるいはECCと呼ばれる冗長データを格納することで信頼性も高めるRAIDというディスクアレイの構成技術が紹介されている。

【0005】つまり、アレイディスクでは、複数のディスクドライブに対して並列に入出力を行うので、データの読み出しあるいは書き込みは高速となり、また、ディスクドライブに障害が発生したときでもパリティと障害ディスクドライブ以外のデータから、障害ディスクドライブのデータを回復することができる。

【0006】より具体的には、上記文献では、データの格納方法によりRAIDレベルを複数に分類している。そのうち、製品で多く使用されるRAIDレベルは、RAID1、RAID3、RAID5である。RAID1はミラーリングであり、2台のディスクに同一データを格納することでディスクの障害に対する信頼性を高めている。読み出し時には、2台のディスクの内どちらか早いほうのディスクから読むことで単一ディスクに比べ高速である。RAID3は、単一データをビットあるいはバイト毎にストライピングし複数のディスクに並列に読み書きすることで、データ転送性能を向上可能である。RAID5は入出力ブロックを単位として複数のディス

## 3

クにストライピングを行うレベルである。

【0007】RAID3では単一入出力要求を小さな単位でストライピングするのに対し、RAID5はそれよりも大きな単位でストライピングを行う。RAID3は単一ユーザの大規模データ入出力を高速化することが主な目的であるが、RAID5は多数ユーザの小規模データ入出力を並列に実行することが主な目的である。従って、RAID3は特に大規模なデータ入出力が要求されるマルチメディアや科学技術計算に用いられる。RAID5は多数ユーザのサービスを行うオンライン・トランザクションデータベース処理に用いられる。一般的なディスクアレイは、前述のキャッシュ制御とディスクの並列動作の両方を取り入れて性能を向上させている。

【0008】一方で、大容量で低コストのディスクアレイの要求が強くなってきている。この場合、単一のディスクアレイコントローラに多数のディスク装置を接続する構成が考えられるが、ディスクアレイ制御装置内の制御プロセッサの性能の限界のために容量は増加するが性能は伸びなくなってしまう問題が発生する。制御プロセッサはホストからの入出力要求を解釈実行するために設けられる。制御しなければならぬディスク装置の台数に比例した処理性能が要求されることから、現状では数台から数十台規模のディスクアレイがほとんどである。

【0009】この問題を解決する方法として、制御装置を階層化する方法が特開平7-44322号公報に述べられている。これは制御装置を階層化することで、各制御装置が制御しなければならない台数を削減することが目的である。例えば、一つの制御装置で5台のディスク装置を制御できるとする。最上位にこの制御装置を1つ設け、その制御装置の複数の出力をそれぞれ異なる制御装置の入力に接続する。こうすることで全体で多数台のディスク装置を制御することが可能になる。

## 【0010】

【発明が解決しようとする課題】特開平7-44322号公報で述べられている階層化されたディスクアレイ制御装置において、キャッシュの適用方法が課題となる。最も簡単な方法は、最上位の制御装置にのみキャッシュを持つことである。しかし、単一の制御装置に接続できるキャッシュ容量は多くないため、多数台のディスク装置に対して小容量キャッシュでは十分な性能が期待できない。そこで、階層化された制御装置の各々にキャッシュを設ける方法が考えられる。この場合、各キャッシュが前述のLRU制御を行うことになる。階層化されたキャッシュの各々で、最も最近アクセスされたデータを残す制御を行うと、キャッシュ間で重複したデータが格納されることになる。

【0011】例えば、制御装置1に制御装置2、3が接続された階層ディスクアレイであり、各制御装置に各々キャッシュが設けられているとする。制御装置2、3に

## 4

は、それぞれ複数のディスク装置が接続されているとする。このとき、制御装置2に接続されたディスクへの読み出し要求が発行された時、制御装置2はディスクからデータを読み込み、LRU制御にしたがって自キャッシュ内にデータを格納する。制御装置2はそのデータを上位である制御装置1へ転送すると、制御装置1も制御装置2と同様に自キャッシュ内にデータを格納する。この場合、上位制御装置のみ格納していればよいデータを複数のキャッシュで重複して格納する問題が発生する。なお、特開平7-44322ではキャッシュの制御方法については述べられていない。

【0012】本発明の目的は、データを複数のキャッシュで重複して格納することを抑制し、キャッシュを有効に使用する手段を提供することにある。

## 【0013】

【課題を解決するための手段】上記目的を達成するために、階層化された制御装置のそれぞれに、自制御装置の階層レベルを格納するメモリを設け、データの読み出し時に前記メモリ内の前記階層レベルを読み出し、最上位制御装置でなければ自キャッシュにはデータを格納しない制御を行うようにする。また、書き込み時にはデータをキャッシュに格納するとともに、パリティ生成に必要なデータが上位キャッシュにないかサーチし、存在すれば上位制御装置から下位制御装置内のキャッシュにデータを転送しパリティ生成を行うようにする。

## 【0014】

【発明の実施の形態】以下、本発明に係る階層ディスクキャッシュ制御方法を図面に示し実施例を参照してさらに詳細に説明する。

【0015】図1は、本発明による階層ディスクキャッシュ制御方法を適用するディスクアレイ制御装置の概略構成図を示したものである。102は、ディスクアレイ制御装置であり、137、138はディスクアレイ制御装置102が入出力を行うディスク装置である。ディスクアレイ制御装置102は、ホストプロセッサ101に接続されている。ホストプロセッサ101からのディスク入出力要求をディスクアレイ制御装置102が受け取り、ディスク装置(137、138)に対して入出力を行う。ディスクアレイ制御装置102は、上位制御装置103と複数の下位制御装置(139、112)から構成されている。複数の下位制御装置(139、112)はディスク装置(137、138)の入出力を行い、上位制御装置103は、ホストプロセッサ101の入出力要求を受けつけ解釈し、複数の下位制御装置(104、112)に対して入出力要求の分配を行う。

【0016】上位制御装置103は、制御プロセッサ106(以下CPUと略す)、キャッシュコントロールメモリ107(以下CCMと略す)、マイクロプログラム108、ホストインタフェース制御部104、上位キャッシュ105、スイッチ109、インタフェース制御部

## 5

(110、111)から構成されている。上位制御装置103の主な機能は、ホストプロセッサ101のディスク入出力要求を解釈し、その結果を複数の下位制御装置(139、112)に分配することである。これらの制御はCPU106によって行われ、この制御の手順や方法が記されたものがマイクロプログラム108と呼ばれるソフトウェアである。

【0017】マイクロプログラム108は通常は半導体メモリに格納されており、CPU106から読み出し・実行が行われる。ホストインタフェース制御部104は、ホストプロセッサ101との接続を制御し、入出力要求を受け付け、データの送受信を行う。スイッチ109は複数の下位制御装置(139、112)に入出力要求を分配する機能を持つ。CPU106からの指示で、入出力要求をどの下位制御装置に分配するのかが決定される。インタフェース制御部(110、111)は下位制御装置(139、112)との接続を制御し、入出力要求の転送やデータの送受信を行う。上位制御装置103には、入出力を高速化するための上位キャッシュ105がある。

【0018】上位キャッシュ105は、ホストプロセッサ101から要求のあったデータを一時的に格納しておくことでディスク装置(137、138)へのアクセスを削減し入出力処理を高速化するために設けられる。CCM107は上位キャッシュの制御方法に関する情報が格納されている。

【0019】下位制御装置(139、112)内は、上位制御装置103との接続を制御するインタフェース制御部(113、125)、下位制御装置(139、112)を制御するCPU(114、126)とマイクロプログラム(116、128)、ディスク装置(137、138)のデータを一時的に格納する下位キャッシュ(124、130)、下位キャッシュ(124、130)に関する制御情報が格納されたCCM(115、127)、ディスクコマンドを分配するスイッチ(123、131)、ディスク装置(137、138)との接続を制御するインタフェース制御(117~121、132~136)、パリティ制御(122、129)から構成されている。

【0020】下位制御装置(139、112)の主な機能は、性能と信頼性を向上させるディスクアレイ制御を行うことである。ディスクアレイには、データやパリティの格納方法により複数の種類があるが、本実施例では、それらのいずれも使用可能である。それらの主な種類は次の通りである。RAID1はミラーリングであり、2台のディスクに同一データを格納することでディスクの障害に対する信頼性を高めている。読み出し時には、2台のディスクの内どちらか早いほうのディスクから読むことで単一ディスクに比べ高速である。RAID3は、単一データをビットあるいはバイト毎にストライ

## 6

ピング(分割)することで、データ転送性能を向上可能である。

【0021】RAID5は入出力ブロックを単位として複数のディスクにストライピングを行うレベルである。RAID3では単入出力要求を小さな単位でストライピングするのに対し、RAID5はそれよりも大きな単位でストライピングを行う。RAID3は単一ユーザの入出力を高速化することが主な目的であるが、RAID5は複数ユーザの入出力を並列に実行することが主な目的である。従って、RAID3は特に大規模なデータ入出力が要求される画像等を処理するマルチメディアや科学技術計算に用いられる。RAID5は多数ユーザのサービスを行うオンライン・トランザクションデータベース処理に用いられる。

【0022】RAID1、3、5ではいずれの場合も、データ格納時に冗長データも同時にディスクに書き込むことで信頼性も向上している。具体的には、RAID1の場合は、同一データを複数のディスクに書き込むことで冗長性を持たせている。またRAID3、5では、データ複数のディスクにストライピングするが、この時、旧データと旧パリティと新データとの排他的論理和を新パリティとしてディスクに書き込む。こうすることで、いずれかのディスクが障害を起こしても、RAID1、3、5いずれの場合も障害を起こしたディスクのデータを回復することが可能である。

【0023】例えば、RAID3において以下のようにデータが格納されているとする。

ディスク1      データ    = 10101010

ディスク2      データ    = 11110000

ディスク3      パリティ= 01011010

この状態でディスク2が障害を起こし読み書きができなくなった場合、ディスク1とディスク3の排他的論理和を演算する。

ディスク1      データ    = 10101010

ディスク3      パリティ= 01011010

論理和                                = 11110000 (ディスク2のデータ)

このように、ディスク2のデータを再現することが可能である。図1のパリティ制御(122、129)はパリティの生成を行う機構である。

【0024】本発明では図1に示すように、キャッシュが上位制御装置103と下位制御装置(139、112)間で階層化されている。キャッシュを階層化することの効果は、要求アクセス時間やコストに応じて、構成を柔軟に変更できることである。一例として、上位のキャッシュ105はコストが高いが高速な半導体メモリで構成し頻繁にアクセスされるデータを保持しておき、下位キャッシュは上位キャッシュに比べ遅いが低コストな半導体メモリで構成するといったことが考えられる。ディスク制御装置のほとんどはキャッシュが単一であった

ために、このような柔軟な構成をとることができなかった。

【0025】本発明の特徴は、キャッシュが階層化されたディスクアレイ制御装置において、上位制御装置103内と複数の下位制御装置(104、112)内にキャッシュ(105、124、130)を設け、互いのキャッシュに格納されたデータの重複を無くす制御を行うことでキャッシュの利用効率を向上することである。

【0026】図2および図3は、本発明における実施例の概要を示している。図2では動作概要を示し、図3では効果を示している。201は上位キャッシュであり、上位キャッシュ201には複数の下位キャッシュ(205、206)が接続されており、階層キャッシュ構造になっている。ホストプロセッサから入力要求が発生した場合の動作を示している。ディスク装置に格納されたデータ204の入力要求が発生した場合、下位キャッシュにデータを格納するかどうかを階層レベルにより決定する。階層レベルとは、ホストプロセッサに最も近いキャッシュを最上位(階層レベル=0)208と定義し、それ以外は階層レベル=1(209、210)である。キャッシュに格納する格納するかどうかは、自キャッシュが最上位キャッシュかどうか判定することで決定する。下位キャッシュ205は、最上位キャッシュではないので、下位キャッシュ205には読み込まれたデータは格納しない212。読み込まれたデータは上位キャッシュ201に転送される。

【0027】ここで、下位キャッシュと同様に最上位キャッシュかどうか判定する(211)。ここでは最上位キャッシュであるため、キャッシュに読み込まれたデータは格納され(202)、そのデータは同時にホストプロセッサに転送される。この時、最上位キャッシュが一杯になった場合、自キャッシュ内のいずれかのデータの追い出し処理を行う。追い出しデータとして選択されたデータ203は、所定の下位キャッシュへ転送される

(207)。下位キャッシュでは、上位から転送されたデータについては、格納するかどうかの判定は行わない。格納するかどうかの判定を行うのは、自キャッシュより上位にデータを転送する時のみであり、上位から転送されたデータは無条件に格納する。これは、上位キャッシュは、最も頻繁に使用されるデータのみを格納し、下位キャッシュには上位キャッシュに格納しきれなかったデータを格納することで、キャッシュのヒット率を向上させるためである。

【0028】図2の制御を行うことで、図3のような効果が得られる。図3は、従来のキャッシュ制御と、本発明の実施例における階層キャッシュの効果を比較した図である。従来のキャッシュ制御は、全てのキャッシュにおいてまったく同じ制御を行っているために、データの重複が生じる。上位キャッシュ1101に12個のデータが格納でき、下位キャッシュ(1102、1103)

にはそれぞれ6個のデータが格納できる場合を示している。

【0029】ホストプロセッサから入力要求が発生すると、上位キャッシュ1101、下位キャッシュ(1102、1103)それぞれで、最も最近アクセスされたデータを残す制御を行う。そのため、キャッシュに格納されたデータが重複するために、キャッシュを無駄に使用することになる。本発明では、上位キャッシュ1104には従来と同様に最も頻繁にアクセスされるデータを残し、下位キャッシュ(1105、1106)には、上位キャッシュ1104に転送したデータは格納しない制御を行っているため、キャッシュに格納されるデータに重複が生じない。そのため、従来は重複データが格納されていた領域は空き領域となり、この領域に新しいデータを格納することができるようになる。

【0030】図4は、本発明における実施例のマイクロプログラム301の構造を示している。上位制御装置103と下位制御装置(139、112)ともに構成は同じである。302はコマンド制御、303は階層キャッシュ制御、304はキャッシュ管理リスト、305はディスク読み込み&先読み制御、306はディスク書き込み&まとめ書き制御、307は階層キャッシュ初期設定制御である。自制御装置より上位の装置からの要求はコマンド制御302で受け付けられ解釈される。その結果、読み込み処理であればディスク読み込み&先読み制御305が呼び出され、書き込み処理であればディスク読み込み&先読み制御306が呼び出される。ディスク読み込み&先読み制御305、ディスク読み込み&先読み制御306内ではそれぞれ階層キャッシュ制御303が呼びだされ自制御装置内のキャッシュ制御が行われる。その際、キャッシュ管理リスト304が使用される。キャッシュ管理リスト304は自キャッシュの使用リスト、空きリストなどが格納されている。階層キャッシュ初期設定制御307は、主に装置起動時に実行され、階層キャッシュの制御に必要な情報を設定する。これらの情報は次に説明する。

【0031】図5は、階層キャッシュの制御に必要な情報が格納されたCCM(107、115、127)のデータ構造を示している。これらの情報は、装置起動時に階層キャッシュ初期設定制御307によって設定される。401は階層レベルフィールドであり、ホストプロセッサに最も近いキャッシュが階層レベル0であり、それ以外は階層レベル1である。キャッシュ制御モード402は、キャッシュの先読みやまとめ書き制御を行うかどうかのフラグが設定されている。キャッシュ使用上限値403とキャッシュ使用下限値404は、キャッシュのあふれや先読み制御をどの時点で行うかの数値が格納されている。キャッシュの使用状況がキャッシュ使用上限値に達したら、下位制御装置またはディスク装置へデータを転送し、未使用領域を増やす。また、キャッシュ

使用下限値を下回ったら下位制御装置またはディスク装置からデータの先読みを行いキャッシュに格納されているデータを増やす。キャッシュ使用リストポインタ405とキャッシュ未使用リストポインタは図4のキャッシュ管理リスト304へのポインタを示しており、キャッシュの確保や開放を行う際に使用される。

【0032】図6は、図4のマクロプログラム301内のコマンド制御302のフローを示している。ステップ501は、上位装置から転送されたディスク入出力コマンド受付処理である。ステップ502はコマンドの判定を行う。ここでは、コマンドが入力要求なのか出力要求なのかが判定され、入力要求（READ）であればステップ503に進み、出力処理（WRITE）であればステップ506に進む。

【0033】ステップ503は、ディスク読み込み&先読み制御を行う。この処理は後で詳細に説明する。ステップ504は、ステップ503で読み込まれたデータに対してキャッシュ制御を行う。ステップ505は上位装置へ終了報告を行う。ステップ506は、上位装置から転送されたデータの階層キャッシュ制御を行う。ステップ507はディスク書き込み&まとめ書き制御を行う。この処理は後で詳細に説明する。ステップ508は、上位装置へ終了報告を行う。

【0034】図7は、図4の階層キャッシュ制御303のフローを示している。ステップ601はコマンド判定処理であり、出力要求か入力要求かを判定する。入力要求であればステップ602に進み出力要求であればステップ609に進む。ステップ602では、CCM（図1の107、115、および、127）内の階層レベルフィールド（図5の401）の読み出しを行う。このフィールドには、最上位キャッシュであれば0が、それ以外のキャッシュであれば1が格納されている。ステップ603は自キャッシュが最上位キャッシュかそうでないかの判定を行う。最上位キャッシュであればステップ604に進み、そうでなければステップ606に進む。ステップ604では、LRU処理を行う。LRU処理とは、キャッシュがフルになった時にどのデータを追い出すかを決定するアルゴリズムであり、キャッシュの管理に一般に用いられている。

【0035】ステップ604では、読み込み要求に対して転送されたデータを最優先にキャッシュに格納するため、もしその時キャッシュがフルになっていた場合、いずれかのデータをキャッシュから追い出す処理を行わなう。ステップ605では、上位装置へデータを転送する。

【0036】ステップ606、607は、自キャッシュが最上位キャッシュではなかった時の処理である。最上位キャッシュでなかった場合は、キャッシングしない制御を行う。これは、さらに上位にキャッシュが存在するため上位のキャッシュに格納されていれば自キャッシュ

にデータを格納する必要がないためである。ステップ606では、上位装置にデータを転送する。ステップ607ではキャッシュ領域を解放する。

【0037】ステップ604、605、606、607により、自キャッシュが最上位キャッシュであればキャッシュに格納し、そうでなければキャッシュには格納しない処理を実現してる。この判定のためにCCM内に階層レベルを示す情報がキャッシュ毎に格納されている。

【0038】ステップ609は書き込み処理の場合に実行され、通常のLRU処理が行われる。この階層キャッシュ制御により、階層化されたキャッシュ構成において、各キャッシュ間に重複するデータをなくすることができ、キャッシュの利用効率を向上することが可能になる。

【0039】図8は、キャッシュ管理リスト304のデータ構造を示している。405は図5におけるキャッシュ使用リストポインタであり、406はキャッシュ未使用リストポインタである。キャッシュ使用リストポインタ405には、自キャッシュ内で使用されている領域がリスト形式で格納されており、各リスト内は次のリストへのポインタ703、使用領域識別子704が格納されている。リストの最後の次のリストへのポインタ703はリストの最後を示す値が格納される。このリストは、参照されるとリストの最も最後にリストチェーンを切り替える制御が行われる。

【0040】これにより、最も最近アクセスされた領域はリストの最も最後におかれ、最も長くアクセスされなかった領域はキャッシュ使用リストポインタ405に最も近いリストにチェーンされている。そのため、キャッシュがフルになり、いずれかの領域を解放しなくてはならなくなったときにはキャッシュ使用リストポインタ405に最も近いリストが選択することでLRU制御を実現している。キャッシュ未使用リストポインタ406に示されるリストは、使用されていないキャッシュ領域がチェーンされており、キャッシュを確保する場合にはリストの先頭か領域が選択される。

【0041】図9は、ディスク読み込み&先読み制御のフローを示している。ステップ801では上位装置より要求されたデータがキャッシュに格納されているかどうかの判定を行う。これは図8のキャッシュ使用リストをサーチすることで実現できる。ステップ802でもしキャッシュにデータが格納されていたら（キャッシュヒット）処理を終了させる。キャッシュに格納されていなければ（キャッシュミスヒット）ステップ803に進む。ステップ803では、キャッシュ領域を確保する。これは、図8のキャッシュ未使用リストポインタが示しているリストを選択することで実現できる。ステップ804ではディスク入力要求を生成し、ディスク装置からデータの入力を行う。ステップ805は先読み制御であるが、この処理はバックグラウンドで実行される。先読み

処理の中断は、図5キャッシュ使用上限値403に達したとき、または次の入力コマンドが発生したときである。

【0042】図10は、ディスク書き込み&まとめ書き制御のフローを示している。ステップ901では、上位装置から転送されたデータを格納するためのキャッシュ領域を確保する。これはキャッシュ未使用リストポイント406に示されるリストを選択することで実現できる。ステップ902では、自キャッシュ内のディスクへ書き込んでいないデータ量がNを超えたかどうかを判定している。もし、Nを超えていたらディスクへの書き込みを実行する。これは、小データの場合に、要求が発生する毎にディスクへの書き込みを行っているため処理時間が増加するため、ある程度まとまった単位でディスクへの書き込みを行うことで性能を向上させるための処理である。ステップ903、904はパリティ生成を行うための処理である。

【0043】前述の通り、ディスクアレイではパリティを生成しディスクに格納することで信頼性を向上させている。パリティの生成には、旧データ、旧パリティ、新データが必要であり、ステップ903ではこれらのデータが上位キャッシュにないかサーチする。これは、本発明はデータを重複しないようにキャッシュ制御を行っているため、最新データが上位装置に存在する可能性があるためである。もし、上位キャッシュ、自キャッシュにパリティ生成に必要なデータが存在しない場合は、下位装置へアクセスしデータを集める。ステップ904では旧データ、旧パリティ、新データの排他的論理和をとり、パリティを生成する。ステップ905では、書き込みデータ（新データ）とパリティを所定のディスクへ書き込むことで処理が終了する。

【0044】図11では、図4における階層キャッシュ初期設定制御307によって、図1におけるCCM(107、115、127)内の階層レベルフィールド(図5の401)にどのような値が格納されるかを示している。これまでの実施例では2階層のキャッシュで説明してきたが、図11に示すように、3階層、あるいはそれ以上でも本発明の効果が得られる。最上位制御装置1001に接続された中間制御装置(1002、1003)、さらに中間制御装置(1002、1003)に接続された最下位制御装置(1004、1005)のような構成の場合、階層レベルフィールドは次のようになる。

【0045】最上位制御装置1001は0、中間制御装置(1002、1003)は1、最下位制御装置(1004、1005)は1に設定することで、各階層間でデータが重複することはなくなる。これまでは装置が正常に動作している場合を述べてきたが、ダイナミックに階

層レベルを変更したほうが良い場合がある。装置あるいはキャッシュに障害が発生した場合である。最上位制御装置1001内のキャッシュが障害を起こした場合は、その次の階層のキャッシュ(図11では中間制御装置1002、1003)が最上位キャッシュ(階層レベルフィールド=0)に設定される。これにより、障害時においても本発明の効果を十分に得ることができるようになる。

#### 【0046】

10 【発明の効果】本発明では、階層化されたキャッシュで構成されるディスクアレイにおいて、下位キャッシュでは、上位にデータを転送する時は自キャッシュにデータを格納しない制御を行う。これにより上位キャッシュは、最も頻繁に使用されるデータのみを格納され、下位キャッシュには上位キャッシュに格納しきれなかったデータを格納されることになるため、キャッシュのヒット率を向上させるためである。

#### 【図面の簡単な説明】

20 【図1】本発明における実施例の全体図を示す図である。

【図2】本発明の実施例における動作概要を示す図である。

【図3】本発明の実施例における効果を示す図である。

【図4】本発明の実施例におけるマイクロプログラム構成を示す図である。

【図5】本発明の実施例におけるキャッシュ制御メモリ(CCM)構造を示す図である。

【図6】本発明の実施例におけるコマンド制御フローを示す図である。

30 【図7】本発明の実施例における階層キャッシュ制御フローを示す図である。

【図8】本発明の実施例におけるキャッシュ管理リスト構造を示す図である。

【図9】本発明の実施例におけるディスク読み込み&先読み制御フローを示す図である。

【図10】本発明の実施例におけるディスク書き込み&制御フローを示す図である。

【図11】本発明の実施例における階層レベルを示す図である。

#### 40 【符号の説明】

101 ホストプロセッサ、

102 ディスクアレイ制御装置、

137、138 ディスク装置、

103 上位制御装置、

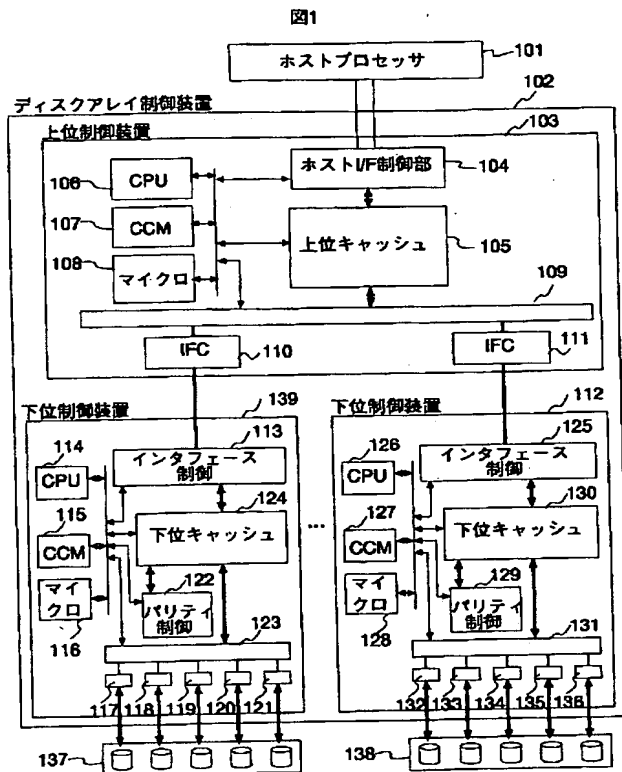
139、112 下位制御装置、

105 上位キャッシュ、

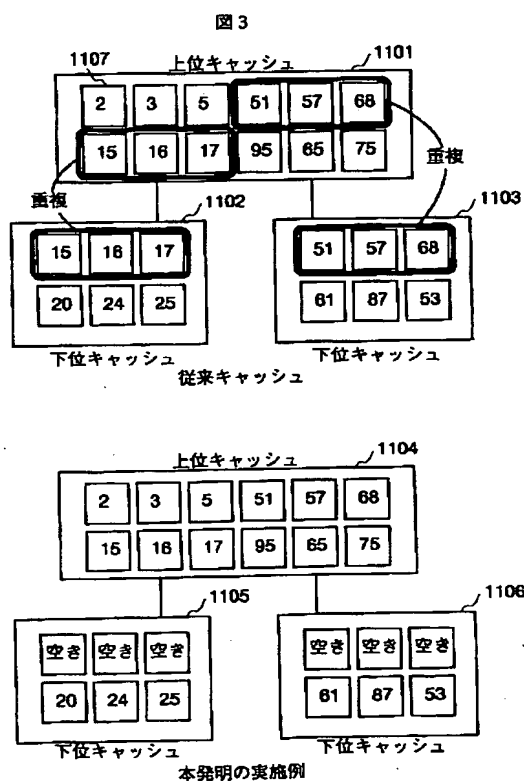
124、130 下位キャッシュ、

107、115、127 キャッシュ制御メモリ。

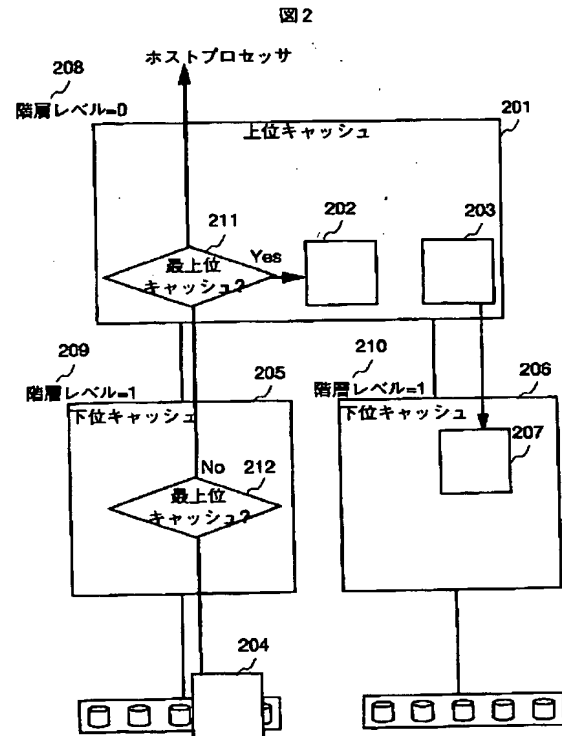
【図1】



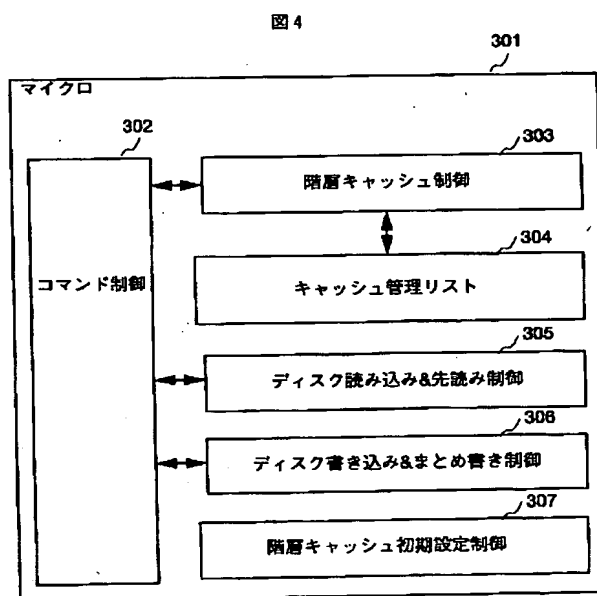
【図3】



【図2】

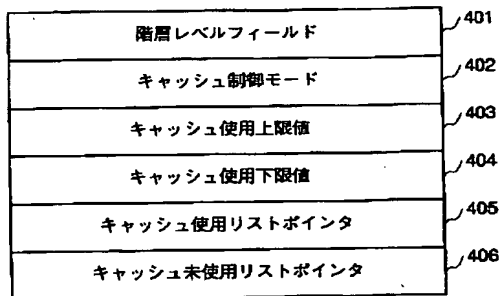


【図4】



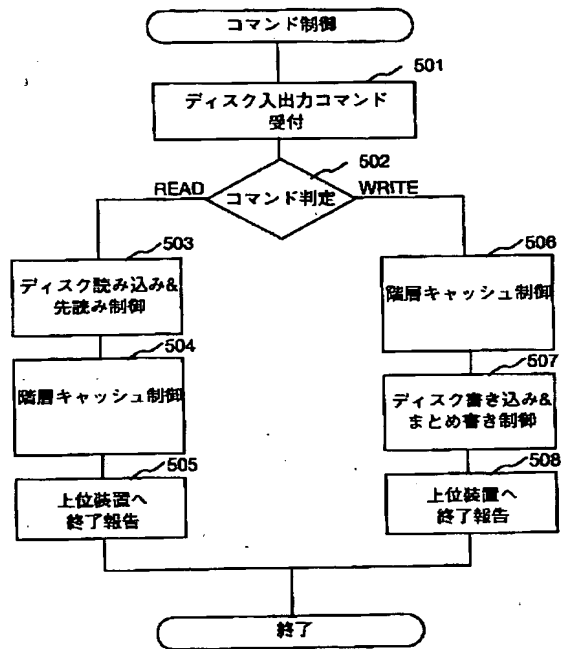
【図5】

図5



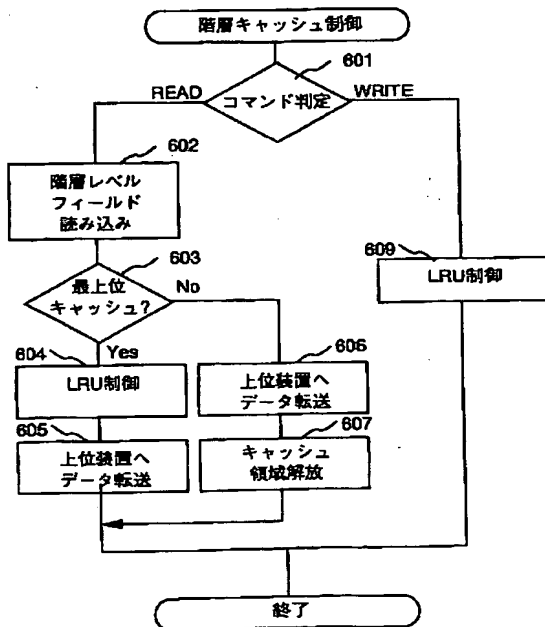
【図6】

図6



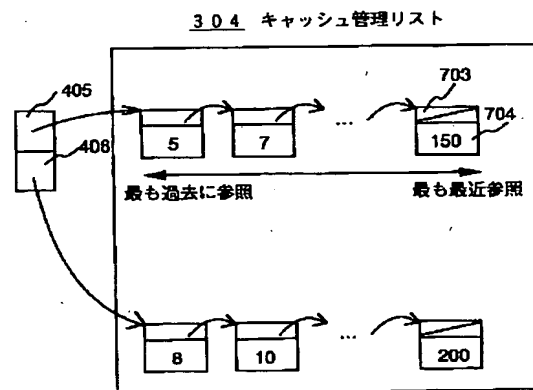
【図7】

図7

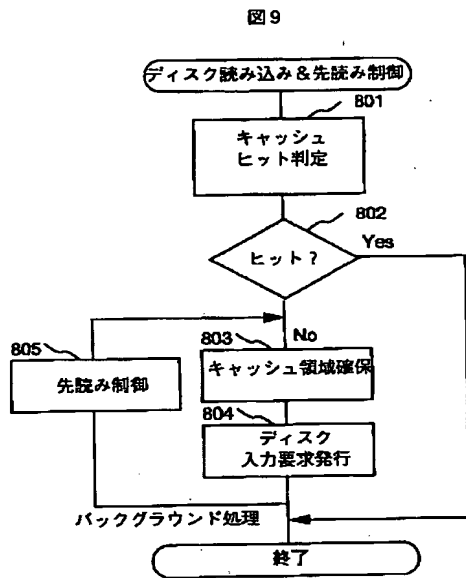


【図8】

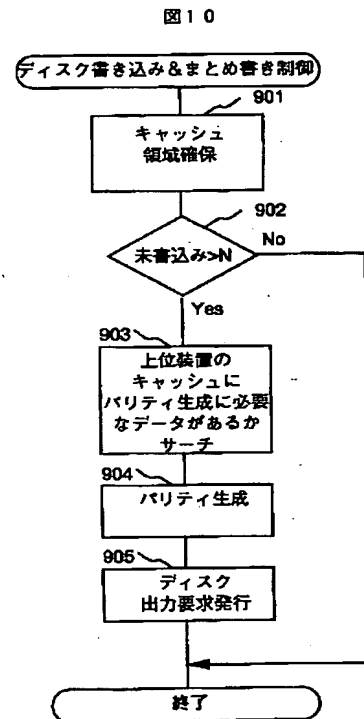
図8



【図9】



【図10】



【図11】

図11

